WHAT IS CLAIMED IS:

1. A computer-implemented method for processing data on a node having a node identifier, the method comprising:

receiving a first data packet on the node from a first direction;

checking a destination identifier of the first data packet;

if the destination identifier of the first data packet does not match the node identifier, storing the first data packet in a first transit buffer for later transmission by the node to another node in the first direction; and

if the destination identifier of the first data packet matches the node identifier,

processing the first data packet on the node to create a first processed data packet; and

storing the first processed data packet in a first local buffer for later transmission by the node to another node in the first direction.

2. The computer-implemented method of claim 1, wherein the method further comprises:

checking a transmission round of the first data packet; and

if the transmission round of the first data packet does not match a transmission round of a previous data packet received on the node from the first direction, changing a first transit buffer round that is associated with the first transit buffer.

3. The computer-implemented method of claim 2, wherein checking a transmission round of the first data packet includes checking a round bit in the first data packet.

4. The computer-implemented method of claim 2, wherein changing a first transit buffer round that is associated with the first transit buffer includes changing a first transit buffer round that is associated with the first transit buffer if one or more data packets are already stored in the first transit buffer.

5. The computer-implemented method of claim 1, wherein the method comprises processing data on a node in a ring network.

6. The computer-implemented method of claim 1, wherein the method further comprises implementing congestion control when the first transit buffer is full.

7. The computer-implemented method of claim 1, wherein the method further comprises:

receiving a second data packet on the node from a second direction;

checking a destination identifier of the second data packet;

if the destination identifier of the second data packet does not match the node identifier, storing the second data packet in a second transit buffer for later transmission by the node to another node in the second direction; and

if the destination identifier of the second data packet matches the node identifier,

processing the second data packet on the node to create a second processed data packet; and

storing the second processed data packet in a second local buffer for later transmission by the node to another node in the second direction.

8. The computer-implemented method of claim 7, wherein the second direction is opposite to the first direction.

9. The computer-implemented method of claim 7, wherein the first data packet and the second data packet have a common size.

10. The computer-implemented method of claim 1, wherein processing the first data packet on the node includes processing the first data packet on the node using a segmentation and reassembly layer.

11. A computer-implemented method for processing data on a node, the method comprising:

determining if a first transit buffer on the node is empty, the first transit buffer capable of holding one or more data packets destined for another node;

if the first transit buffer is empty, transmitting in a first direction a data packet stored in a first local buffer, the first local buffer capable of holding one or more data packets originating from the node;

if the first transit buffer is not empty,

transmitting in the first direction one or more data packets stored in the first transit buffer if a first transmission condition is satisfied; and

transmitting in the first direction a data packet stored in the first local buffer if the first transmission condition is not satisfied.

12. The computer-implemented method of claim 11, wherein transmitting in a first direction a data packet stored in a first local buffer if the first transit buffer is empty includes changing a transmission round associated with the transmitted data packet.

13. The computer-implemented method of claim 12, wherein changing a transmission round associated with the transmitted data packet includes changing a round bit in the transmitted data packet.

14. The computer-implemented method of claim 11, wherein data packets transmitted in the first direction have a common size.

15. The computer-implemented method of claim 11, wherein transmitting in a first direction a data packet stored in a first local buffer if the first transit buffer is empty includes transmitting one or more data packets stored in the first local buffer.

16. The computer-implemented method of claim 11, wherein transmitting in the first direction a data packet stored in the first local buffer if the first transmission condition is not satisfied includes changing a transmission round associated with the transmitted data packet.

17. The computer-implemented method of claim 11, wherein transmitting in the first direction a data packet stored in the first local buffer if the first transmission condition is not satisfied includes transmitting in the first direction a data packet stored in the first local buffer if the first local buffer contains one or more data packets.

18. The computer-implemented method of claim 11, wherein transmitting in the first direction one or more data packets stored in the first transit buffer if a first transmission condition is satisfied includes transmitting in the first direction one or more data packets stored in the first transit buffer if

a previous data packet transmitted in the first direction was empty, or

the previous data packet transmitted in first direction had been stored in the first local buffer, or

a transmission round associated with the previous data packet transmitted in first direction matches a transmission round of a first data packet stored in the first transit buffer.

19. The computer-implemented method of claim 11, wherein the method comprises processing data on a node in a ring network.

20. The computer-implemented method of claim 11, wherein the method further comprises:

determining if a second transit buffer on the node is empty, the second transit buffer capable of holding one or more data packets destined for another node;

if the second transit buffer is empty, transmitting in a second direction a data packet stored in a second local buffer, the second local buffer capable of holding one or more data packets originating from the node;

if the second transit buffer is not empty,

transmitting in the second direction one or more data packets stored in the second transit buffer if a second transmission condition is satisfied; and

transmitting in the second direction a data packet stored in the second local buffer if the second transmission condition is not satisfied.

21. The computer-implemented method of claim 20, wherein the second direction is opposite to the first direction.

22. A computer-implemented method for processing data between nodes in a distributed network, the method comprising:

maintaining a set of local buffers and a set of transit buffers for each node in the distributed network, the set of local buffers for a given node being used for storing data originating at the given node, and the set of transit buffers for the given node being used for storing data received by the given node but destined for another node in the distributed network; and

using the local buffers and the transit buffers to process data between the nodes in processing cycles, wherein each node is capable of receiving data from another node and storing this data in one of its transit buffers during one processing cycle, and wherein each node is capable of transmitting data from one of its local buffers and from one of its transit buffers to another node during one processing cycle.

23. The computer-implemented method of claim 22, wherein each node is capable of receiving data from another node, storing this data in one of its transit buffers, and changing a transit buffer round associated with the transit buffer containing the stored data if a transmission round of the received data does not match a transmission round of previously received data.

24. The computer-implemented method of claim 22, wherein each node is capable of transmitting data from one of its local buffers and from one of its transit buffers to another node and changing a transmission round associated with the transmitted data.

25. The computer-implemented method of claim 22, wherein the method comprises processing data between nodes in a ring network.

26. The computer-implemented method of claim 22, wherein:

maintaining a set of local buffers and a set of transit buffers for each node in the distributed network includes maintaining at least two local buffers and at least two transit buffers for each node in the distributed network, such that each local buffer and each transit buffer is associated with a particular direction of data transmission.

27. A computer-readable medium having computer-executable instructions contained therein for performing a method, the method comprising:

receiving a first data packet on a node from a first direction, the node having a node identifier;

checking a destination identifier of the first data packet;

if the destination identifier of the first data packet does not match the node identifier, storing the first data packet in a first transit buffer; and

if the destination identifier of the first data packet matches the node identifier,

processing the first data packet on the node to create a first processed data packet; and

storing the first processed data packet in a first local buffer.

28. A computer-readable medium having computer-executable instructions contained therein for performing a method, the method comprising:

determining if a first transit buffer on a node is empty, the first transit buffer capable of holding one or more data packets destined for another node;

if the first transit buffer is empty, transmitting in a first direction a data packet stored in a first local buffer, the first local buffer capable of holding one or more data packets originating from the node;

if the first transit buffer is not empty,

transmitting in the first direction one or more data packets stored in the first transit buffer if a first transmission condition is satisfied; and

transmitting in the first direction a data packet stored in the first local buffer if the first transmission condition is not satisfied.

29. A distributed network, comprising:

a plurality of nodes that each have a set of local buffers and a set of transit buffers, the set of local buffers for each node being used for storing data originating at that node, and the set of transit buffers for each node being used for storing data received by that node but destined for another node in the distributed network;

wherein each node is capable of receiving data from another node and storing this data in one of its transit buffers during one processing cycle; and

wherein each node is capable of transmitting data from one of its local buffers and from one of its transit buffers to another node during one processing cycle.